



10/539493
10/539493

17 JUN 2005

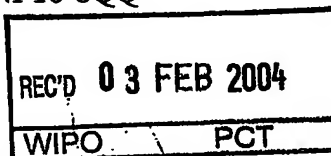


INVESTOR IN PEOPLE

The Patent Office
Concept House
Cardiff Road
Newport
South Wales
NP10 8QQ

PRIORITY DOCUMENT

SUBMITTED OR TRANSMITTED IN
COMPLIANCE WITH RULE 17.1(a) OR (b)

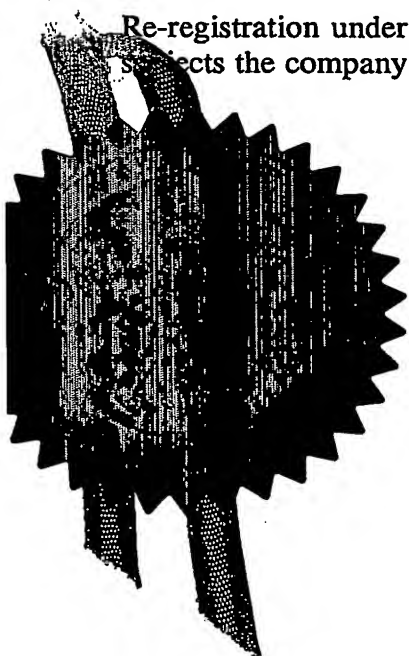


I, the undersigned, being an officer duly authorised in accordance with Section 74(1) and (4) of the Deregulation & Contracting Out Act 1994, to sign and issue certificates on behalf of the Comptroller-General, hereby certify that annexed hereto is a true copy of the documents as originally filed in connection with the patent application identified therein.

In accordance with the Patents (Companies Re-registration) Rules 1982, if a company named in this certificate and any accompanying documents has re-registered under the Companies Act 1980 with the same name as that with which it was registered immediately before re-registration save for the substitution as, or inclusion as, the last part of the name of the words "public limited company" or their equivalents in Welsh, references to the name of the company in this certificate and any accompanying documents shall be treated as references to the name with which it is so re-registered.

In accordance with the rules, the words "public limited company" may be replaced by p.l.c., plc, P.L.C. or PLC.

Re-registration under the Companies Act does not constitute a new legal entity but merely affects the company to certain additional company law rules.



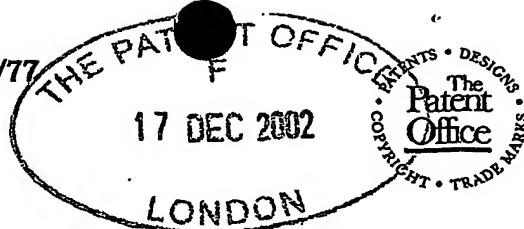
Signed

Dated

22 January 2004

BEST AVAILABLE COPY

An Executive Agency of the Department of Trade and Industry



18DEC02 E771671-12 D02718
P01/7700/0.00-0229368.6

Request for grant of a patent

(See the notes on the back of this form. You can also get an explanatory leaflet from the Patent Office to help you fill in this form)

17 DEC 2002

The Patent Office

Cardiff Road
Newport
South Wales
NP10 8QQ

1. Your reference	SSA/P8474GBA		
2. Patent application number (The Patent Office will fill in this part)	0229368.6		
3. Full name, address and postcode of the or of each applicant (underline all surnames)	Aspex Technology Limited Denmark House Denmark Street High Wycombe, Buckinghamshire HP11 2ER		
Patents ADP number (if you know it) If the applicant is a corporate body, give the country/state of its incorporation	8529026001 England and Wales		
4. Title of the invention	Improvements Relating to Parallel Data Processing		
5. Name of your agent (if you have one)	David Keltie Associates		
"Address for service" in the United Kingdom to which all correspondence should be sent (including the postcode)	Fleet Place House 2 Fleet Place London EC4M 7ET United Kingdom		
Patents ADP number (if you know it)	040145020006 ✓ 4014502006		
6. If you are declaring priority from one or more earlier patent applications, give the country and the date of filing of the or of each of these earlier applications and (if you know it) the or each application number	Country	Priority application number (if you know it)	Date of filing (day / month / year)
7. If this application is divided or otherwise derived from an earlier UK application, give the number and the filing date of the earlier application	Number of earlier application	Date of filing (day / month / year)	
8. Is a statement of inventorship and of right to grant of a patent required in support of this request? (Answer 'Yes' if: a) any applicant named in part 3 is not an inventor, or b) there is an inventor who is not named as an applicant, or c) any named applicant is a corporate body. See note (d))	YES		

Patents Form 1/77

9. Enter the number of sheets for any of the following items you are filing with this form. Do not count copies of the same document

Continuation sheets of this form

Description 12

Claim(s)

Abstract

Drawing(s)

10. If you are also filing any of the following, state how many against each item.

Priority documents

Translations of priority documents

Statement of inventorship and right to grant of a patent (*Patents Form 7/77*)

Request for preliminary examination and search (*Patents Form 9/77*)

Request for substantive examination (*Patents Form 10/77*)

Any other documents
(please specify)

11.

I/We request the grant of a patent on the basis of this application.

David Keltie Associates

Signature

Date

David Keltie Associates

17 December 2002

12. Name and daytime telephone number of person to contact in the United Kingdom

Ted Chwu
020 7329 8888

Warning

After an application for a patent has been filed, the Comptroller of the Patent Office will consider whether publication or communication of the invention should be prohibited or restricted under Section 22 of the Patents Act 1977. You will be informed if it is necessary to prohibit or restrict your invention in this way. Furthermore, if you live in the United Kingdom, Section 23 of the Patents Act 1977 stops you from applying for a patent abroad without first getting written permission from the Patent Office unless an application has been filed at least 6 weeks beforehand in the United Kingdom for a patent for the same invention and either no direction prohibiting publication or communication has been given, or any such direction has been revoked.

Notes

- If you need help to fill in this form or you have any questions, please contact the Patent Office on 08459 500505.*
- Write your answers in capital letters using black ink or you may type them.*
- If there is not enough space for all the relevant details on any part of this form, please continue on a separate sheet of paper and write "see continuation sheet" in the relevant part(s). Any continuation sheet should be attached to this form.*
- If you have answered 'Yes' Patents Form 7/77 will need to be filed.*
- Once you have filled in the form you must remember to sign and date it.*
- For details of the fee and ways to pay please contact the Patent Office.*

Improvements Relating to Parallel Data Processing

Introduction

The present invention concerns improvements relating to parallel data processing. The invention is intended for use in digital computers. More specifically, digital computers that have a Single Instruction Multiple Data (SIMD) data processor. The purpose of the invention is to accelerate the inter-processor communications for a specific sub-class of operation known as alternation.

Efficient inter-processor communications in SIMD data parallel processors is paramount to performance. The requirement is to provide efficient hardware support for binary 'divide-and-conquer' algorithms, whereby the multitude of processors in a data parallel architecture can be sub-divided into even/odd subsets in an efficient manner.

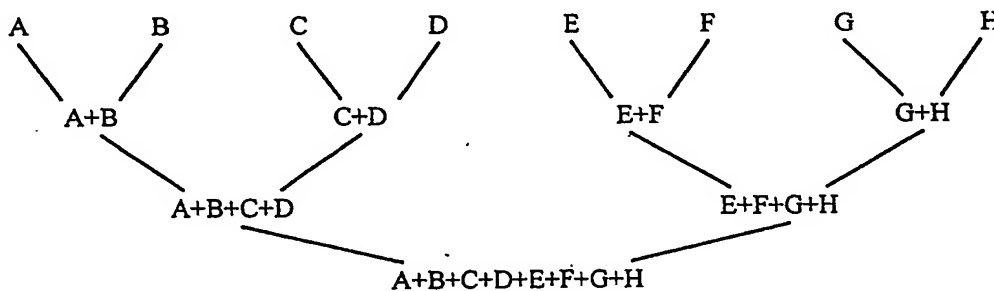


Figure 1 Divide-and-conquer principal of operation

Figure 1 illustrates this principal, showing the summation of 8 numbers by successively adding even-odd pairs in $\log_2 N$ steps, where $N=8$.

The major attributes of such a hardware network, known as the *alternation network* are:

- Provision of hardware augmentation for divide-and-conquer algorithms.
- Transparent operation across chip boundaries in modular multi-chip solutions with a minimum of external hardware support.
- Operation to complete in a minimum number of clock cycles - ideally one - reasonably independent of the number of devices participating in the operation.

Although the invention is general in its application early instances of it have been manufactured for use with the Aspex ASP (Associative String Processor) data parallel processor. The ASP is a SIMD data processor that in typical configurations operates on 1024 to 65536 data items in parallel. Some of the invention's detail is specific to this data processor. The major attributes of the ASP are:

- 1152 processing elements on a single device 12.0mm x 12.0mm in size.
- 100MHz clock speed.

Future versions of this processor will incorporate up to 8192 processing elements.

One of the features of the ASP processor is the string (one-dimensional) topology of the processor array, wherein the processing elements are connected end-to-end. This facilitates some of the features of the network described in this invention.

Prior Art

Simple solution based on EXOR gates

The simplest solution for an alternate operation is to provide a linear string of EXOR (Exclusive-OR) gates, one per processor, connected so that the control signal is local to the processor, and the communication signal propagates to the next processor.

Figure 2 shows such a solution. The data processor's instruction generates a status signal from each processor called a match, where the match signal from the n th processor is termed M_n . The application of the match signals causes the generation of a signal which propagates between processors, swapping state whenever it encounters a condition where a local match (M_n) is true. By combining the propagation term (LAO_n) in true or complement form with the original match signal, then two new match states can be generated, termed M_{n-even} and M_{n-odd} .

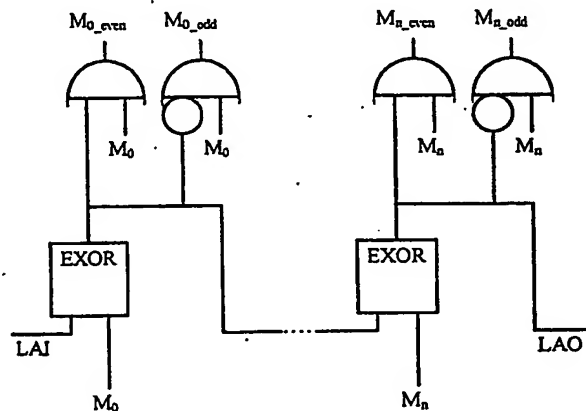


Figure 2 Basic hardware alternation network

The versatility of this solution is that it allows the even and odd match sets to be easily generated, offering application flexibility. Its major disadvantage is the poor speed of the solution, since the propagating signal is limited by the gate delay of a single EXOR gate and the number of processors in the string.

An Example

An example of an alternation network in the VASP-256 data parallel processor comprises a network of 256 EXOR gates, one per processor. Execution of an alternate instruction requires approximately 11 clock cycles (at 50MHz) to completely generate an even-odd partition of M_0 to M_{255} .

Invention

Introduction

The invention is a new alternation network architecture that achieves the functionality of generating alternating match sets at a vastly improved speed over the current art. It does this by generating an initial alternate locally within a segment, then generating a correction term based on the forward propagation of the output of the local segment EXOR chains. This output provides information about whether the total number of matches within a segment are even or odd. This correction term is forward propagated through the network, utilising a hierarchy of bypass levels, which further accelerate the propagation of information. In a subsequent cycle, the correction term is utilised to generate a new input to the local segment, which then serves to cause the segment to maintain or reverse initial alternate to achieve the correct final state.

The novelty of the invention is in:

- The generation of a pair of match vectors, representing alternating sets of even and odd matching processors .
- The use of a circuit for multiplying the rate at which the final alternating sets are generated.
- The capability to partition the alternating sets into local subsets.
- The capability to control the *inject* condition at the *head* of each subset, thereby controlling the generation of the alternating sets such that the *leftmost* matching APE may belong to the even or odd sets.
- The capability to exploit the topology of the network to implement *fault-tolerance* with no further logic.

Network Terminology and Definitions

Segment	A collection of APEs, typically 16.
Block	A collection of segments, typically 8.
Chip	A collection of blocks.

The Alternating Communications Network Detail

Figure 3 shows the basic modular Alternation Network cell.

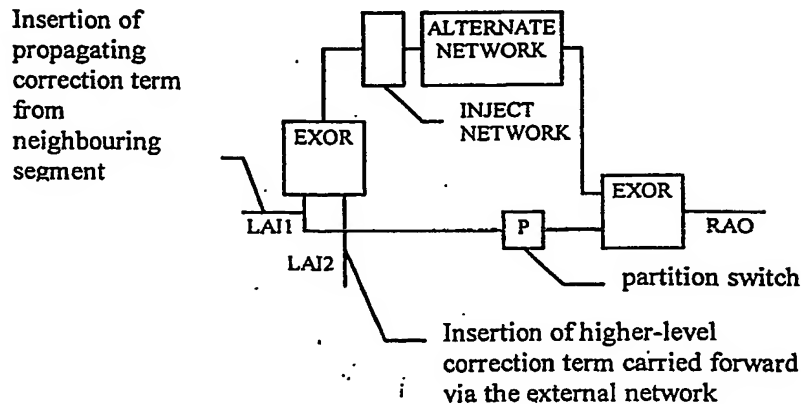


Figure 3 Alternation network low-level module

The module labelled **ALTERNATE NETWORK** may be considered to correspond to the network shown in Figure 2. Typically this portion of the network comprises only a few stages (processing elements), since propagation through this network must be carried out twice,

1. as a pre-condition to the propagation of correction terms, and
2. as a post-condition to apply the correction terms for the final generation of the even-odd sets.

Consequently this sub-network must be kept as short as possible, consistent with the requirements of the floorplan.

The module labelled 'P' (partition switch) is a simple switch function controlled by a signal P, where s denotes the segment identifier. For example s=0 is considered to be the *leftmost* segment. If this signal is TRUE then the carry forward of the correction term via the partition switch is disabled.

If this signal is TRUE then communication into this segment is also disabled, a task handled by the module labelled **INJECT NETWORK**, which controls the insertion of a propagating signal into the leftmost end of the **ALTERNATE NETWORK**, as well as responding to the partition control signal (P_s) to regulate input to the network.

Inject Network Detail

Figure 4 is a block diagram of the inject network.

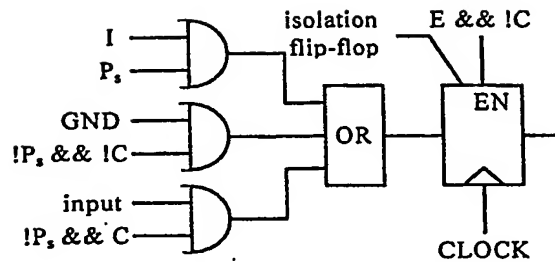


Figure 4 Inject network detail

The signal 'I' represents the global inject state. The inject state may be used to determine whether the leftmost participating processor in a subnet belongs to the even or odd sets. This inject state is only applied if the propagate term for the given segment (P_i) is TRUE, resulting in this subnet being partitioned from segments to its *left*. The global inject state then overrides all forward propagating correction terms into the network input and unconditionally becomes the input to the ALTERNATE NETWORK.

The signal C is true during the *correction* phase of the network operation, when communication network will be generating a carry forward correction term into the network *input* which will be used to modify the initial state of the ALTERNATE NETWORK (provided that P_i is FALSE). The initial state of the ALTERNATE NETWORK is used as the basis of generating the correction terms and propagating them forward through the network, hence the correction term is isolated from the ALTERNATE NETWORK by the *isolation flip-flop*.

Once the correction term has been generated and is stable, it will be clocked through to the ALTERNATE NETWORK during the evaluate phase of the network operation, when the correction term will be enabled through the isolation flip-flop.

When the signal C is false, then the isolation flip-flop will be pre(set) or pre(reset) depending upon the state of the inject input (I) and the partition flag (P_i).

Block-level Hierarchy

The module shown in Figure 3 can now be readily cascaded to form a higher-order network. For example, Figure 5 shows a number of identical networks of the type shown in Figure 3 cascaded to form a *block* network.

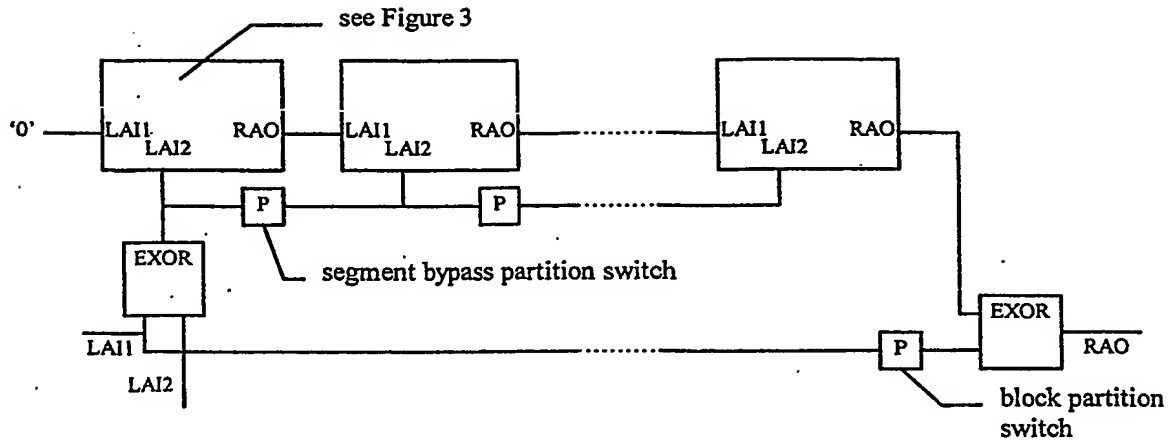


Figure 5 Block-level hierarchical network topology

The block partition switch is controlled by a signal (P_b) which is the logical combination of individual segment partition (P_s) control signals. In this way, only if all segment partition switches within the span of a block are closed is the forward propagation of a correction term enabled.

Such a hierarchical arrangement can be extended indefinitely, for example by cascading block level networks (Figure 5), plus the addition of the corresponding external network (Figure 6). In this example it is assumed that a chip comprises an array of blocks, as per the definitions above.

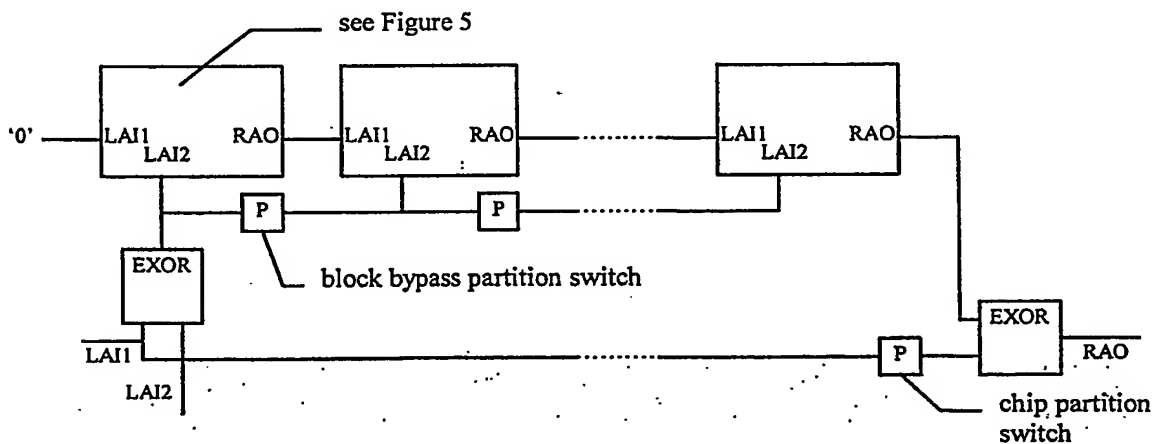


Figure 6 Block-level hierarchical network topology

Inter-chip Network Topology

The hierarchical network topology can be readily extended within the constraints of :

- the target speed of the correction phase and
- the floorplan topology of the integrated circuit.

Therefore the hierarchy within a given chip may be extended to comprise more than the two levels described above.

Ultimately however, the communications signals must propagate off-chip, supporting the goal of transparent chip-chip communication, as well as supporting the continuation of the network logical hierarchy with a minimum of external logic.

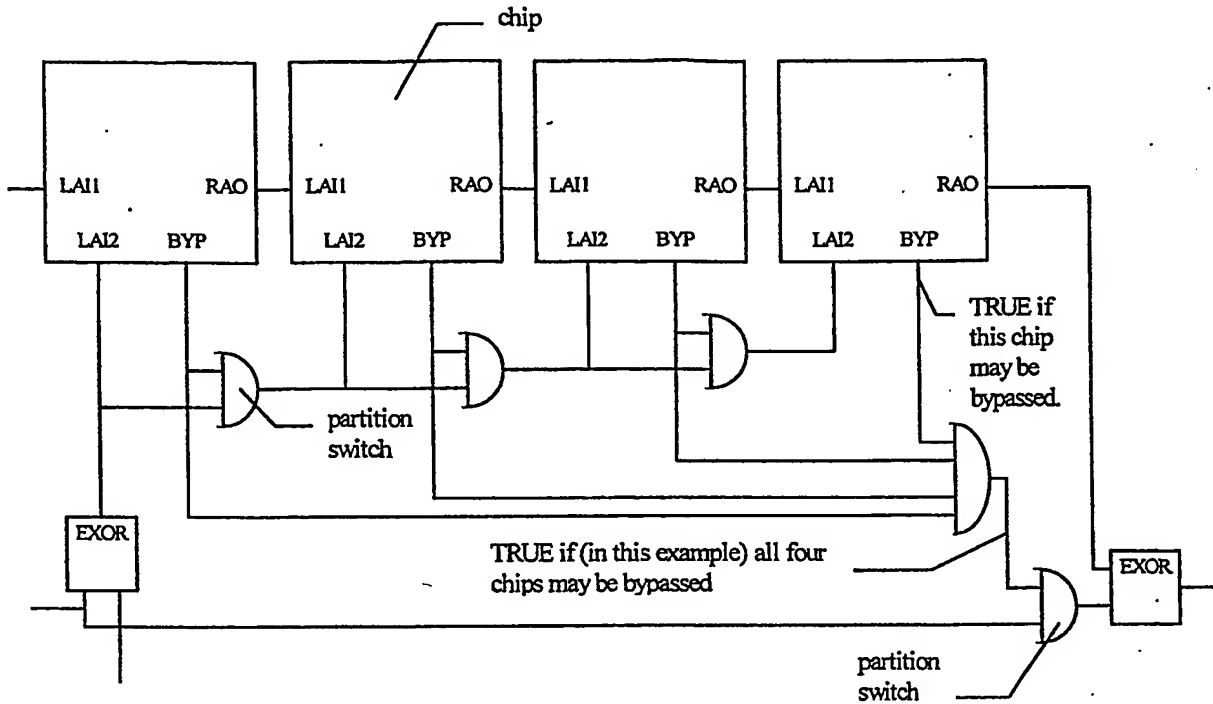


Figure 7 Multi-chip network topology

Figure 7 illustrates how the network hierarchy may be readily extended across multiple chips. In this example a new signal (labelled BYP in the diagram) has been introduced, which is assumed to be TRUE if the chip *may* be bypassed (i.e. no open switch exists anywhere within the device).

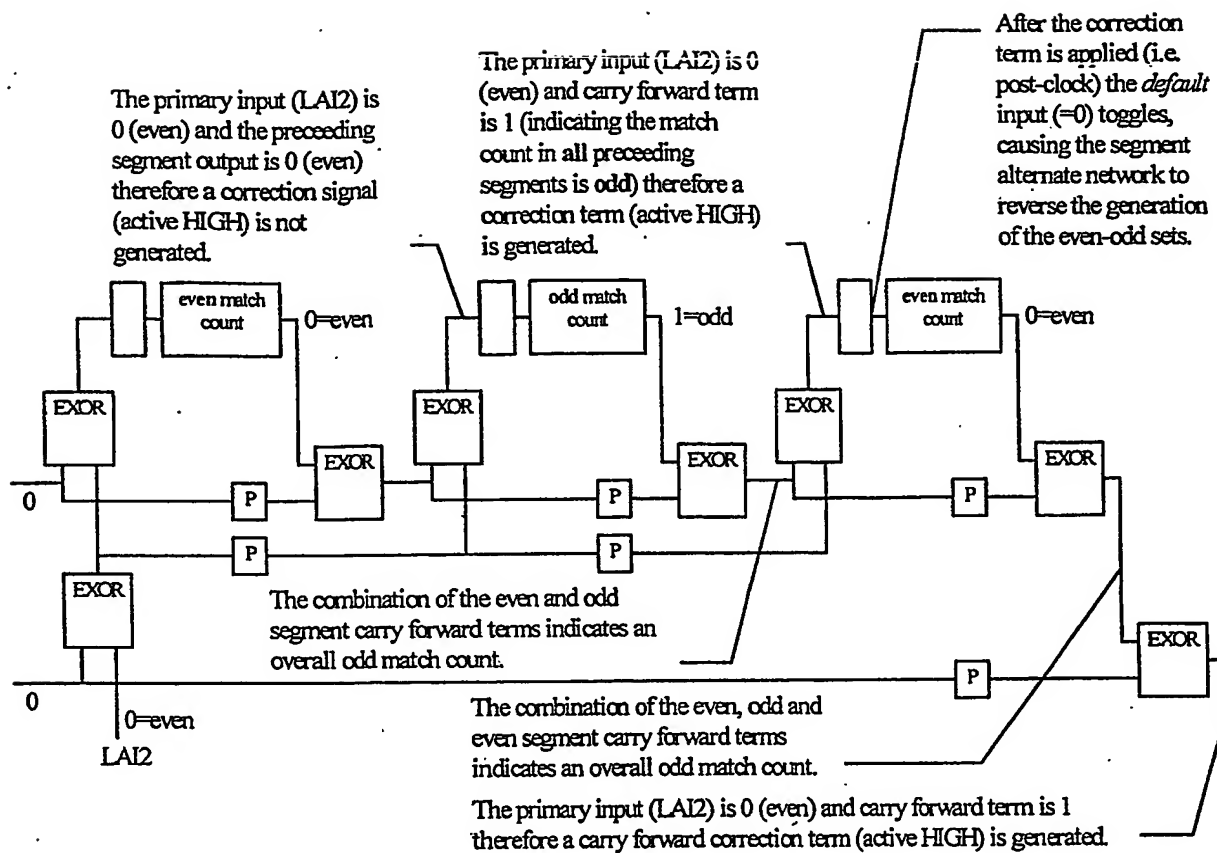


Figure 8 Principal of operation

Principal of Operation

The principal of operation of the network is illustrated in Figure 8, which shows a representative *block* comprising three *segments*. In this example, no partition flags are set, therefore the network is continuous and unbroken along its length. The initial state of the individual segments is evaluated in parallel (assuming a default inject input set FALSE), resulting in indications of even and odd match counts. These are then combined by the carry forward network to produce the correction terms. The role of the EXOR networks becomes clear, since the carry forward of pairs of odd correction terms (FALSE, FALSE) or pairs of even correction terms (TRUE, TRUE) always results in the generation and propagation of an even (TRUE) carry forward term. Similarly, the application of, say, an even and an odd correction term (TRUE, FALSE) results in the propagation of an odd correction term (FALSE).

The duration of generation and propagation of the correction terms is determined by the internal modular hierarchy within the network.

The final stable correction terms are applied on a subsequent clock edge, when the isolation flip-flop in the inject network is clocked. Following the application of the correction term, a sufficient interval must be allowed for the correction term to propagate through the individual segment networks to achieve the final even-odd split.

Whenever an operation is extended (such as the generation and propagation of the correction terms) this is implemented in the context of a synchronous processor device by issuing repeated instructions according to a deterministic (compiled) algorithm.

Instruction Repetition

A typical usage of the network requires a sequence of one or more of the correction data processor instructions, followed by a evaluation processor instruction and a single propagation instruction. Control logic within the chip instruction unit allows the length of this sequence to be deterministic.

An Example

Consider the following example. The instruction sequence that needs to be fed to the data processor for a particular application is:

`{{A}RepeatSlot(4), B }`

where `{..} RepeatSlot(N)` means repeat the instruction inside the braces N times.

Example ASP alternation Code

Consider the following example. Count participating APEs and write the count into the data register in the rightmost APE in the string.

The pseudo-code for this is:

```
identify participating APEs;
A = 0;
FOR X = LSB TO MSB DO
    alternate participating APEs into even and odd sets;
    IF rightmost APEs is in odd set THEN
        A[rightmost][X]=1; (* Xth bit of rightmost APE is set *)
        mask odds and reassign evens as new participating APEs
    END
```

The actual ASP code to perform this task is given below:

```
Initialise
{ {InitWriteSerial(every,sOpen,dX,dX,0,0,aw{ab2.d0,ab1.d0},NoRead)},
  {ExecWrite(bmD,NoCsoOpt)}};

Start of loop, initialise for counter and set intra and inter chip inject
inputs to TRUE
{Set(agbInject),
 For(sfsz,1),
  {SegmentInject(1)},
  {NoOp()}};

Finding participating processors and generate alternate even-odd subsets
{ {InitSearchOrAddSerial(dX,dX,0,0,aw{ab2.d0,ab0.d1})},
  {ExecSearchOrAddSerial(bmD,trAlt,NotForceZero),
```

```

RepeatSlot(AlternateSlots));};

Set intra-chip inject inputs FALSE
{ {SegmentInject(0)},
  {NoOp()}};

Set inter-chip inject input FALSE
{Clear(agbInject),
 {InitWriteSerial(at,slOpen,dX,dX,0,0,aw{ab1.d1},NoRead)},
 {ExecWrite(bmD,NoCsoOpt)}};

Establish that rightmost participating APE is odd
{ {InitWriteTernary(anl,slOpen,tf0,XT,ScalarBus,aw{})},
  {NoOp()}};
{ {InitWriteSerial(all,slOpen,dX,dX,0,0,aw{ab2.d1},NoRead)},
  RepeatSlot(ActivateSlots)},
 {ExecWrite(bmD,NoCsoOpt)}};
{ {InitSearchOrAddSerial(dX,dX,0,0,aw{ab0.d1,ab1.d0,ab2.d0})},
 {ExecSearchOrAddSerial(bmD,tr1,NotForceZero)}};
{ {InitWriteTernary(anr,slOpen,tf0,XT,ScalarBus,aw{})},
  {NoOp()}};

Write result LSB
{Init(sfal,sfbal),
 {InitWriteSerial(aerf,slOpen,d1,dX,sfal,0,aw{ab2.d1},NoRead)},
 RepeatSlot(ActivateSlots)},
 {},
 Inc(sfal));
{ {},
 {ExecWrite(bmD,NoCsoOpt)}};

Mask current odd set from participating in next cycle of alternates
{ {InitWriteSerial(every,slOpen,dX,dX,0,0,aw{ab1.d0},NoRead)},
 {ExecWrite(bmD,NoCsoOpt)}},
 EndFor();

```

The evaluation of this task using the invention would result in completion of this application in 240 clock cycles instead of 304 clock cycles using the network shown in Figure 2, which represents the prior art. The effect is far more significant in the next planned ASP device, which will incorporate 4096 APEs. Here completion of the algorithm will require some 400 clock cycles, compared to some 2500 clock cycles without the invention.

Extension of the concept to support yield-enhancement

The network topology allows the subnet (at segment, block, chip etc. levels) to be readily bypassed in the event of a defect which results in faulty operation. For example, Figure 9 shows the addition of an isolation gate in the ALTERNATE NETWORK output, forcing the output to a null state in the event of a fault in the network. By forcing the partition switch (P) unconditionally *closed* we may exploiting existing hardware to allow any forward propagating correction terms to unconditionally bypass the segment (or block, chip etc.).

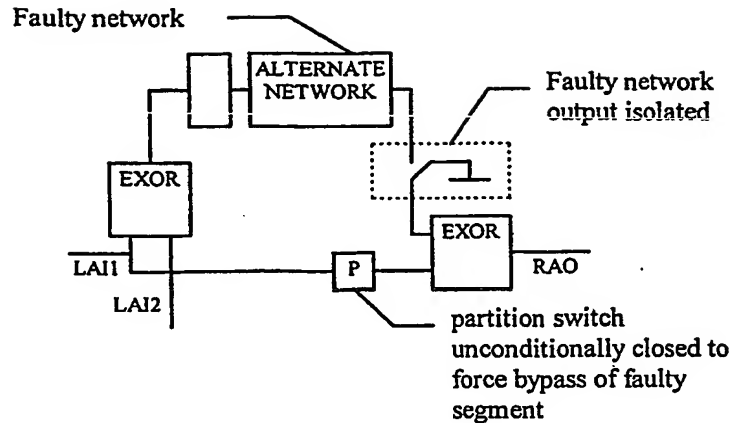


Figure 9 Network amendments for fault-tolerant bypassing

An Implementation

An implementation of the invention, in the VASP-4096/TX chip has completed design. It completes a full alternation across 4096 data parallel processor in a simulated interval of 24ns - or 5 clock cycles at 160MHz. It is implemented in a single 0.25 μ m silicon process.

Benefits

The main benefits of the invention are:

- **Speed.** The ability of complete an alternating communication operation across 4096 APEs in <30ns.
- **Size.** The network at the first level of the propagation hierarchy is essentially identical to existing examples. The network required to generate the correction terms can be easily accommodated within the existing floorplan without significant area impact.
- **Cost and Reliability.** The network is amenable to implementing fault-tolerant bypassing of failed segments and blocks with only minor amendments, thereby adhering to compatibility with the rest of the communication network which exploits similar fault-tolerance. This results in higher yields and significant cost savings in volume manufactured parts.

These benefits open a number of new application areas. The application of this network to VASP processors allows the architecture to implement divide-and-conquer algorithms on parallel data sets much more efficiently. The creation of even-odd sets in an associative processor must be able to effectively exploit the non-deterministic nature of the processor, which allows participating elements of the initial active set to be distributed across a large number of physical processors with indeterminate physical/logical locations.

The overall device cost is not only reduced by the small size of the network but also:

- it conforms to the general communication link topology,

- it is amenable to fault-tolerance through selective bypassing, thereby opening the possibility of significant savings in chip manufacturing cost, and
- it accommodates controllable inject states with a minimum of hardware.

**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- ☐ BLACK BORDERS
- ☐ IMAGE CUT OFF AT TOP, BOTTOM OR SIDES
- ☐ FADED TEXT OR DRAWING
- ☒ BLURRED OR ILLEGIBLE TEXT OR DRAWING
- ☐ SKEWED/SLANTED IMAGES
- ☐ COLOR OR BLACK AND WHITE PHOTOGRAPHS
- ☐ GRAY SCALE DOCUMENTS
- ☐ LINES OR MARKS ON ORIGINAL DOCUMENT
- ☐ REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY
- ☐ OTHER: _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.